MPEG 104th Meeting, Incheon, Korea

JCT3V-D0301 AHG 10 : Overview of Algorithmic Intrinsic Complexity Measures

G. G. Lee (NCKU)

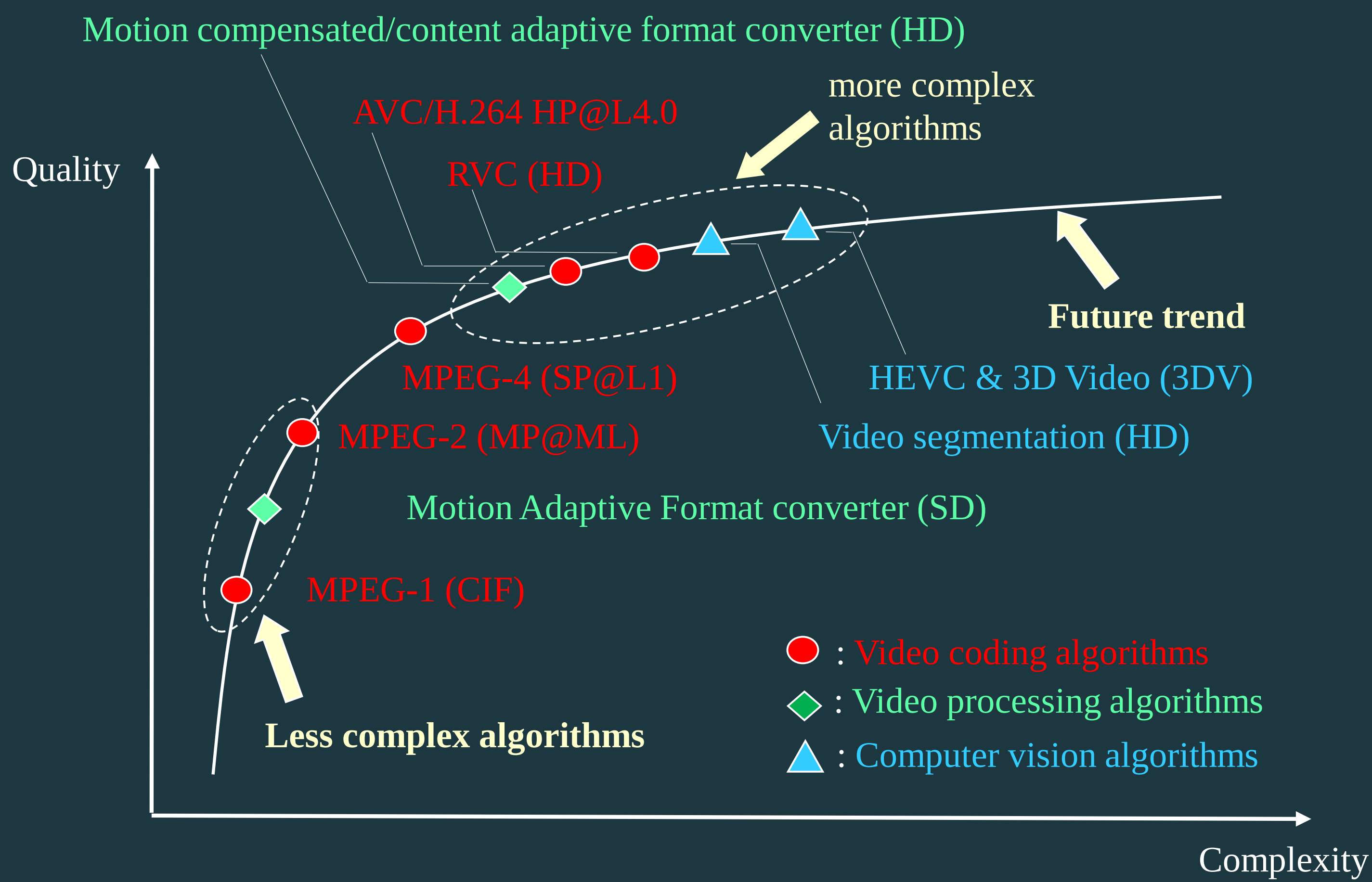
T. Ikai (Sharp)

K. Rapaka (Qualcomm)

D. Rusanovskyy (Nokia)

Table of Contents

Qualitative complexity of visual computing algorithms in MPEG & JVT/JCT

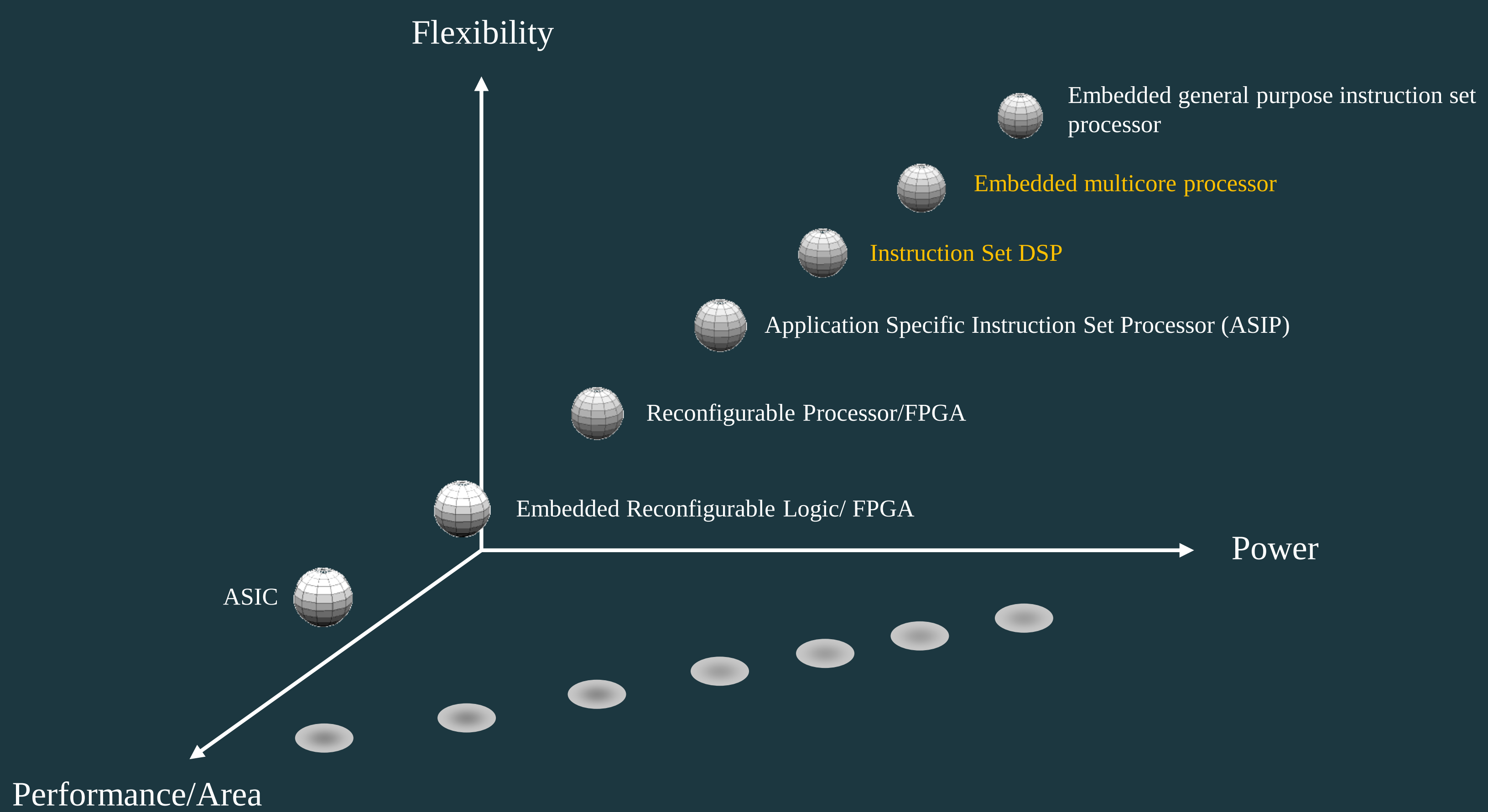


High level complexity analysis of algorithms

Trend of increasing complexity of algorithms described qualitatively :

1. More usage of temporal or motion information
   - Video Coding
     - MPEG-1 has frame mode motion estimation (ME), MPEG-2 added field mode
     - MPEG-4 and H.264/AVC uses more temporal information like direct mode
     - AVC/H.264 uses more frames for inter mode and also variable block sizes (VBS) ME
2. Enhancement of content adaptivity
   - Video Coding
     - MPEG-4 adapts to reference blocks to the left and top
     - H.264/AVC has Context adaptive Variable Length Decoder (VLD), CABAC
     - also VBS to fit different content of the image sequences
3. More views and depth with also larger data sizes and higher range or data precision
   - Resolution went from QCIF to CIF, then from SD to HD and more and more
   - Data accuracy went from 8 bits to 10 bits and beyond like 14 or 16(?)
   - Now HEVC goes beyond HD and even more

Spectrum of platforms or architectures



Generic design scenarios

Scenario I:

Dataflow

Fully specified algorithm → Freely developed platform or architecture

Scenario II:

Dataflow

Freely specified algorithm → Already existing platform or architecture

Scenario III:

Dataflow

Freely specified algorithm → Freely developed platform or architecture

Algorithmic intrinsic complexity analysis

Complexity metrics (measures)

- The complexity metrics should be:
  - *Transparent to hardware and software* implementations
  - *Indicative of the software, hardware or system architecture requirements*
  - *Known early in the design phase* to avoid changes
  - ***Platform independent***
- The complexity metrics to look at are:
  - Number of operations
  - Data storage requirement
  - Data transfer rate
  - Degree of parallelism

Complexity metrics: number of operation

- This metric calculate the numbers of each type of operation
  - Division
  - Multiplication
  - Addition/subtraction
  - Logical operations
  - Shift
- Operations with constant input and variable input should be differentiated to provide high accuracy
  - $X + Y$ vs. $X + 5$ (X and Y are variables)
  - $X \times Y$ vs. $X \times 5$ (X and Y are variables)
- In addition, the ***precision*** of each operand should be taken into account, since it can significantly influence complexity

Complexity metrics: data storage requirement

- Data storage requirement is transparent to hardware/software but depends on the processing order or dataflow of algorithms.

- It is important to distinguish between:
  - the data storage which is ***intrinsic*** to the algorithm
  - memory configuration with design constraints

- Data storage requirement of algorithms can be obtained by analyzing the ***lifetime*** of the required input data.

Complexity metric: data transfer rate

- It is important to distinguish between algorithmic intrinsic and implementation specific information
- Average bandwidth
  - The amount of data transferred in one second
  - Algorithm-intrinsic metric
- Instantaneous or peak bandwidth
  - The amount of data transferred during a short time unit
  - Architecture-dependent metric
    - Types of data transaction
    - Memory hierarchy
    - Data alignment in memory and bus protocol
    - Datapath architecture.

Adjacency matrix and Laplacian matrix

Graph

Adjacency matrix **A**

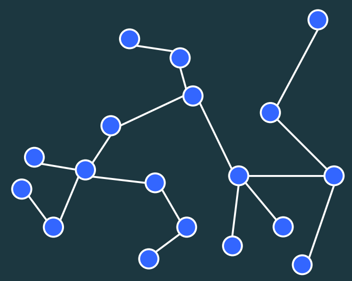
$$\mathbf{A}(\mathrm{i},\ \mathrm{j}) = \begin{cases} 1 & \text{if vertex}_i \text{ and vertix}_j \text{ are adjacent to each other} \\ 0 & \text{otherwise} \end{cases} \Rightarrow \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Laplacian matrix **L=D-A**, where **D** is **a** diagonal matrix where the diagonal elements represents the number of edges connected to that node.

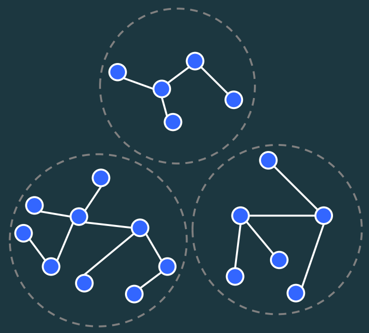
$$\mathbf{L}(\mathrm{i},\ \mathrm{j}) = \begin{cases} \mathbf{D}(\mathrm{i},\ \mathrm{j}) & \text{if } i = j \\ \text{-}1 & \text{if } i \neq j \text{ and vertex}_i \text{ is adjacent to vertix}_j \\ 0 & \text{otherwise} \end{cases} \Rightarrow \mathbf{L} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

Properties of the spectrum of Laplacian matrix

Connected graph



A graph composed of three connected components



- The Laplacian matrix of a connected graph has only one eigenvalue = 0.
- If a graph is composed of several connected components, the spectrum of the graph is the union of the spectra of its connected components.
- The number of connected components in the graph is equal to the multiplicities of the 0 eigenvalue of the Laplacian matrix of the graph.

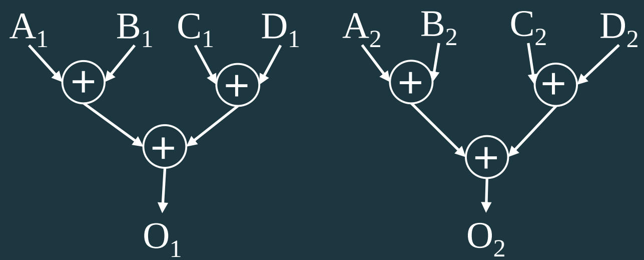
Extraction of parallelism

Algorithm

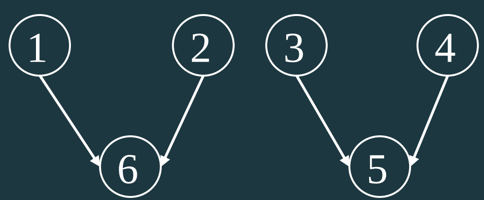
$O_1=A_1+B_1+C_1+D_1$

$O_2=A_2+B_2+C_2+D_2$

Dataflow diagram



Causation graph



Laplacian matrix

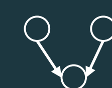
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Spectrum

Eigenvalue: 0, 0, 1, 1, 3, 3

Eigenvector: $\begin{bmatrix}1\\1\\0\\0\\0\\1\end{bmatrix} \begin{bmatrix}0\\0\\1\\1\\1\\0\end{bmatrix} \begin{bmatrix}0\\0\\-1\\1\\0\\0\end{bmatrix} \begin{bmatrix}1\\-1\\0\\0\\0\\0\end{bmatrix} \begin{bmatrix}0\\0\\1\\1\\-2\\0\end{bmatrix} \begin{bmatrix}-1\\-1\\0\\0\\0\\2\end{bmatrix}$

Parallelism

2 × 

(Homogeneous)

Advantages of this approach

- This approach provides a systematic way to quantify the parallelism of algorithms.
- The spectrum of Laplacian matrix of a graph is an invariant of the graph matrix regardless of the ways or orders in which the vertices are labeled .
- Matrices are easier to handle

Graph	1 — 2 — 3	2 — 1 — 3	1 — 3 — 2
Laplacian matrix	$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}$
Eigenvalue	0 1 3	0 1 3	0 1 3
Eigenvector	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}$

Some Notes

- All four complexity measures should be evaluated during complexity assessment
- The data granularity used in the dataflow model during complexity assessment will result in different implementations